Edges

```
(ns opencv3.threshold
  (:require
    [opencv3.core :refer :all]
    [opencv3.utils :as u]))

(def neko
  (->
  "resources/images/cat.jpg"
  imread
  (u/resize-by 0.2)))
(u/mat-view neko)
```

```
(def factor 6)
(def background (new-mat))
(def work (clone neko))

(dotimes [_ factor] (pyr-down! work))
(bilateral-filter work background 9 9 7)
(dotimes [_ factor] (pyr-up! background))
(resize! background (new-size (.cols neko) (.rows neko)))
(u/mat-view background)
```

```
(def
  c
  (-> neko
  clone
  (blur! (new-size 3 3))
  (cvt-color! COLOR_BGR2GRAY)
  (canny! 300.0 100.0 3 true)
  (bitwise-not!)
  (cvt-color! COLOR_GRAY2BGR)  ))
(u/mat-view c)
```

```
(let [result (new-mat) ]
  (bitwise-and background c result)
  (u/mat-view result))
```

```
(def t
(-> neko
  clone
  (blur! (new-size 3 3))
  (cvt-color! COLOR_BGR2GRAY)
  (threshold! 150 255 THRESH_BINARY_INV)
  (cvt-color! COLOR_GRAY2BGR)  ))

(let [result (new-mat) ]
  (bitwise-and background t result)
  (u/mat-view result))
```